IBM confidential

TR 22.256

Technical Report

A Computerized Approach to Manufacturing Planning

R. Sadofsky

IBM Systems Manufacturing Division
EAST FISHKILL

TR 22.256

IBM

A Computerized Approach to Manufacturing Planning

R. Sadofsky

May 2, 1966

ABSTRACT

This paper describes a computer program for long- and short-term manufacturing planning. It it designed to evaluate the effects of changes to manufacturing variables such as mix, requirements, yields, lead times, capacities and products.

Locator Terms for the IBM Subject Index

Planning, manufacturing
Capacity, plant
Production, planning
05 Computer Application

R. Sadofsky, Dept. 933, Manufacturing Resources and Long-Term Planning

Prepared by: Technical Publications, IBM Components Division East Fishkill, Hopewell Junction, N. Y.

Printed in U. S. A.

INTRODUCTION

The computer program described in this report was devised as an aid in manufacturing planning for the East Fishkill facility. The facility manufactures many products simultaneously, each product having its own set of yields, lead times, and operation sequence. In addition, products may go through the same operation more than once (aside from rework). All operational capacities are therefore dependent upon product mix if the schedule is held constant. Also, since there are different yield losses and lead times for different product types, the schedule is also dependent upon mix if the total quantities desired to stock is held constant.

In the situation just described, capacity planning under one set of groundrules (mix, requirements, yields, lead times, etc.) is a tedious task since a schedule must be worked out operation-by-operation (taking into account yields, lead times, mix requirements, etc.). Then, these throughputs must be compared operation-by-operation to the planned capacities. A change in the groundrules requires a complete regeneration of schedules, comparison to capacities, etc. Also, if it is desired that results to date be taken into account, the task gets complicated further. In most instances, it takes so long to evaluate changes to the groundrules that all timeliness is lost before the task is completed. Once data on each operation is determined, it is also desirable to know what is the sequence of limiting operations on a common basis (how does each operation restrict how much product can get to stock in relation to each other).

Basically, then, the purpose of this program is to manipulate data to evaluate changes to any or all of the variables mentioned and to answer questions relative to these variables. The program is written in FORTRAN IV and requires 32K words of memory. Refer to Appendix A for the complete program listing and Appendix B for data formats.

THE MANUFACTURING PROCESS

The manufacturing process upon which the program is based is the manufacture of transistors and diodes. At present, there are eleven distinct product types. However, the following description applies in general to all types.

General Process Description and Flow

Flow	Approximate No. of Operations	Description
1	15	A silicon crystal is grown. It is ground, inspected, and sliced into wafers (approximately thirty-five wafers per inch). The wafer is then lapped and etched. An epitaxial layer is deposited upon the wafer, and an insulating layer is deposited over the epitaxial wafer. The wafer is then inspected.
2	6	A light-sensitive solution is put on the water, and, with a photographic mask, component patterns are transferred to the wafer. The patterns are then etched on wafer and then the wafer is cleaned and inspected.
3	5	Boron is deposited on the etched areas of the wafer and oxidation is performed to diffuse it into the wafer.
4		Step 2 is repeated.
5	6	Phosphorous is deposited on the etched areas of the wafer and an oxidation is performed to diffuse the phosphorous into the wafer.
	10	A layer of gold is then deposited on the back side of the wafer and is driven into the wafer through a furnace operation.
6	8	Step 2 is repeated and the wafer is probed to determine its electrical characteristics.
7	7	A layer of eluminum is evaporated upon the wafer through a mask in order to form a low resistance contact between the silicon and the external metallurgy.
8	6	Another type of light-sensitive solution is put onthe wafer; and steps similar to 2 are performed.
9	9	A layer of glass is deposited on the wafer.
10	6	Step 8 is repeated.
11	8	Cr-Cu-Au and Pb-Sn are evaporated through a mask into the holes formed by step 10.
12	5	Copper balls are put into the holes and soldered to the metals evaporated in the holes through the previous operation to form a contact.
13	5	The wafers are diced into small chips, each one an independent transistor or diode pair.
14	5	Damaged chips are picked off the wafer and the chips are separated (one wafer contains approximately 1,000 chips). Each chip is then independently tested for its electrical characteristics.
15	5	The chips are inspected visually for mechanical defects and the product is complete.
16		The product goes to stock.

The products described require approximately two and one-half working months to manufacture, i.e., go from step 1 to step 16. Losses can occur throughout the process, each product independent of the others. The overall yield loss amounts to approximately 75 percent of the product started at crystal pulling. Since the technology involved in the process is recent, many of the equipment and process steps are continually changing and at best can be "good estimates" where the future is concerned. Demand for the products and the mix of products is also constantly changing.

GENERAL APPROACH

With the facilities and products just described, there is an enormous amount of data. And, any portion of the data involved is subject to change. Any computerized approach to planning must concentrate on minimizing the amount of data. Otherwise, the time required to prepare changes to data for a particular run would restrict the use of the program.

The approach taken with regard to data in the present program is as follows: A basic set of data is read in for the first time-period. From this set, a second set is created. Then, changes to this second set are read in on an exception basis in the second time period. A third set is created from the second set and changes are read in the third time period. The process continues in the same fashion until memory restrictions set in. At this point, there is enough data in memory to make a run. As we move along in time, the next set of changes are read in the bank of original data (the first set is destroyed). This process, therefore, allows a particular run to cover any length of time periods desired and minimizes the amount of data preparation since only exceptions to previous data are read in.

The following data inputs are used by the program:

1. Actual yields which have already occurred
2. Projected yields which are expected to occur
3. Actual throughputs which have already occurred
4. Planned capacities
5. Engineering yield contingencies
6. Usable chips per wafer
7. Mix of product already started
8. Mix of product desired in the future.

The time base about which the program operates is the time when it is expected that product will reach stock. Then, using lead time data, it is determined what portion of the data in memory is applicable to the product coming off the line at the point in time that is under consideration. In other words, the computer reaches back through time in the memory bank, picking up data which is applicable to the product coming off the line at a later date. This concept must be kept in mind when reading the output in order to interpret it correctly and make it meaningful.

SIMPLIFIED EXAMPLES

For the examples that follow, an explanation of terms used and where one is situated in time is in order. (Refer to the general process description and flow.)

OPERATION	SECTION OF FLOW
Crystal Pulling	1
Base Mask	2
Third Mask	6
Aluminum Evaporation	7
Dicing	13
Test	14
Stock	16

SYMBOL	DESCRIPTION
TA	Transistor type A
TB	Transistor type B
TUA	Transistor type UA

EXAMPLE 1: NOVEMBER OUTPUT TO STOCK

Suppose we want to have a certain amount of product (TA, TB, etc.) go to stock in November. Assume that it is the end of August. August events are known, while September data and on are projections.

Using the lead time data, the initial step is to find out where in the line each type of product will have to be in order to come off the line as a finished device in November. For instances, in order to complete a TA device in November, we may have to be at the following points in the line:

Crystal Pulling ↓	August	Based on actual lead times
Base Mask ↓	August	
Third Mask ↓	September	Based on planned lead times
Dicing ↓	October	
Stock	November	

Once the location of each product type is established for a given output to stock, the projected and actual yields for the appropriate months are located.

Crystal Pull ↓	August	Based on actual yields
Base Mask ↓	August	
Third Mask ↓	September	
Dicing ↓	October	Based on planned yields
Stock	November	

These projected and actual yields are accumulated from the end of the line to the beginning of the line in order to determine how many chips (wafers) are needed at any point in the line to give one good chip (wafer) to stock.

For example: Suppose there are three yield points in the line for a TA wafer:

(1) Base mask = 0.8 (August actual)

(2) Aluminum evaporation = 0.5 (October planned)

(3) Final test = 0.25 (November planned)

Thus, at final test in November, for every 4 good chips into final test in November, we will get one good chip into stock in November. For every 8 good chips into aluminum evaporation in October, we will get one good chip into stock in November, not only because of the yield expected at aluminum evaporation in October but also due to the yield expected at final test in November. For every 10 good chips into base mask in August, which were actually processed in August, we will get one good chip into stock in November because of the yield which was actually experienced in August at base mask and the yields which are expected at aluminum evaporation and final test in October and November, respectively.

This yield is then factored by any engineering yield contingency which has been estimated and by the number of chips per wafer for each product type at all operations where chips are still in wafer form in order to be able to compare everything in terms of chips.

If a certain mix is desired at the end of the line (if the product has not yet been started at base mask), this factor is applied against the factors above in order to determine the weight of the above factors in terms of product load. For example, suppose that 10 TA chips were required at base mask in order to to get 1 good TA chip to stock and 30 TUA chips were required in order to get 1 good TUA chip. If a mix of 90% TA and 10% TUA was desired at stock it would be necessary to start 12 chips (90% x 10 + 10% x 30 = 12) at base mask.

If, on the other hand, the mix has already been determined at base mask as the result of actual yields which have already occurred and those which are expected to occur further down the line, what product mix will come off the end of the line has already been determined. Therefore, the mix at the end of the line can be determined by the yields and base mask mix, and the factor can then be applied as it was in the previous case.

For a given month's output to stock, these factors are then applied against actual throughputs if they have already occurred and against planned capacities of the operations have not yet occurred. Therefore, the capacity in terms of finished devices can be determined for each operation (planned), and the expected output to stock based on those operations which already have occurred can be determined. This allows the determination of the number of devices that the line can produce based on what is already in it and where the devices will be bottlenecked as they proceed toward stock. This allows one to see where the queues are occurring in the line and where we can expect them to occur in the future.

EXAMPLE 2: SEPTEMBER OUTPUT TO STOCK (REFER TO FIG. 1)

In this example, the computations would be of the following nature :

(a) Based on the actual throughput which occurred in July through A, if this product is allowed to flow through the line following the actual lead times, yields and mix at A which have already occurred for July and August for A and operations between A and B, and based on the lead times and yields which are expected to occur in September near the end of the line, X_1 amount of product will flow into stock during September.

(b) Based on the actual throughput which occurred in August through B, if the product is allowed to flow through the line following the actual lead times, yields and mix at B which have already occurred for August and based on the planned yields and lead times that are expected to occur near the end of the line, X_2 amount of product will flow into stock during September.

(c) Based on the planned capacity at operation C, if this operation is fully manned, it has the capability of supplying to stock in September a quantity X_3.

If $X_3 > X_1 > X_2$, then

(1) C in September is capable of handling what is in the line.

(2) An inventory $X_1 - X_2$ has built up during the month of August in front of B.

(3) If the inventory can be reduced or eliminated through overtime at B, the maximum the line will produce in September is X_1; if not, the maximum is X_2.

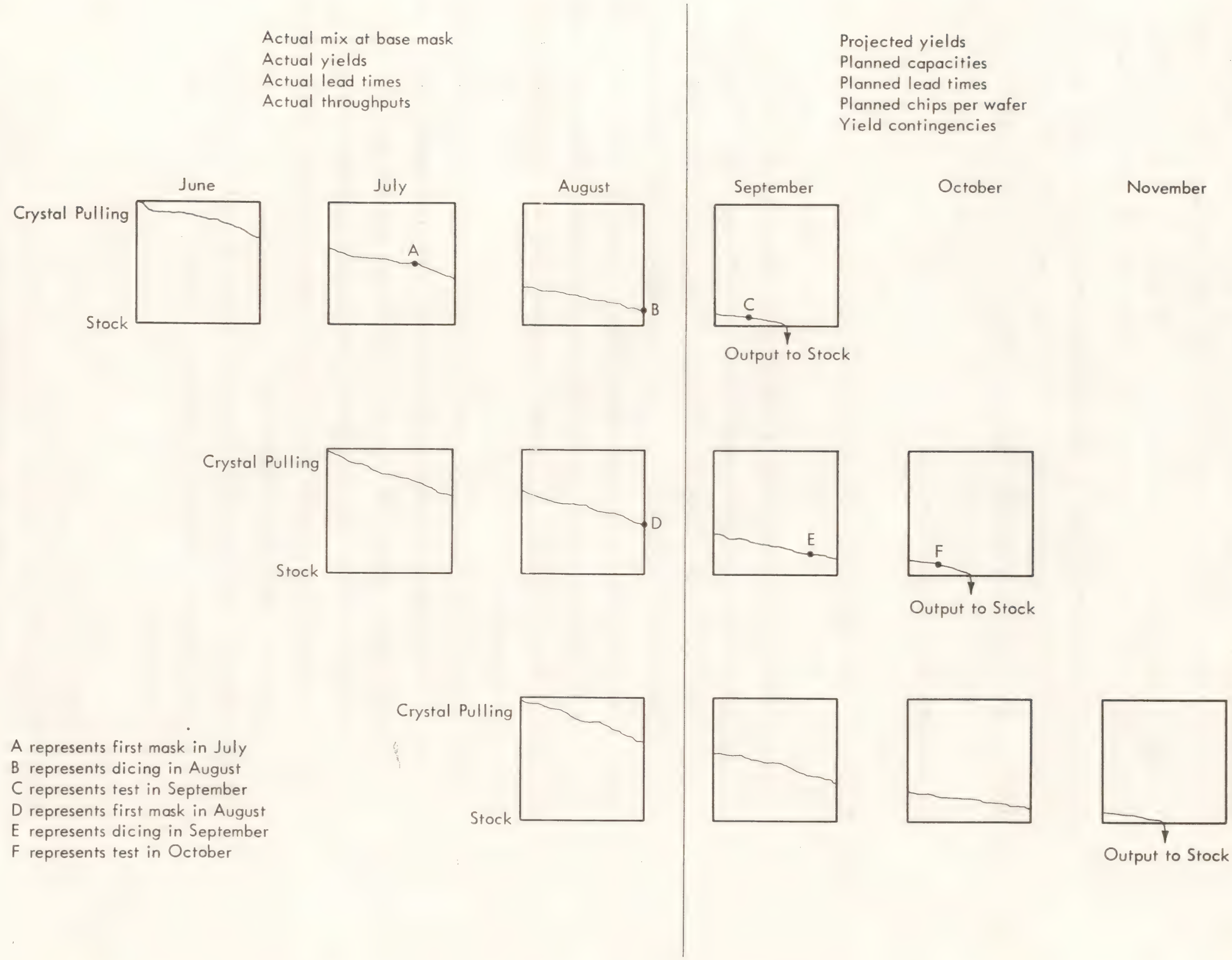

Fig. 1. Graphic representation of mechanics of program.

Conversely, if $X_1 > X_2 > X_3$, then

(1) An inventory $X_1 - X_2$ will have built up during the month of August in front of X_2 .

(2) If C is not altered in September, an inventory $X_2 - X_3$ will form in front of C during the month of September.

(A typical program output is shown in Fig. 2.)

PROGRAM APPLICATIONS

- Vendoring out of Products

By changing the planned-mix input cards, we can alter the percentages between products, and eliminate products that are planned to be vendored out. Thus, after making this change, we can refer to the output column "equivalent finished device capacity" and determine what quantities of devices can be manufactured under the new mix conditions for each time period.

By the same token, the vendor's conditions can also be set up on a separate program. Then, if a need for additional products being sent to this vendor is under evaluation, the variables associated with the new products under consideration can be inputed and, along with his present products, the vendor's capability under the new set of conditions can be evaluated.

- What are the best products to produce at a particular location?

By changing the planned-mix input cards, we can determine what amount of product will flow to stock under different sets of conditions. This may be done under planned or actual conditions to determine a best situation of allocation of products between plants, maximizing production and taking the guesswork out of distributing the workload.

- What is the plant capacity?

By choosing the operation which has the lowest equivalent finished device capacity for each time period, we know what restricts or will restrict the quantities of output for each time period. This operation is then the plant capacity for that time period. If the quantities required are greater than those quantities to which the plant is restricted, we know exactly how much the operation must be increased in order to meet the requirements and by what point in time it must be accomplished. Then the process may continue with the second most limiting operation, third most limiting operation, etc.

Operation Name		Planned Capacity	Equivalent Fin Device Capacity	Oct Output to Stock		Product Impacts Capacity during Month Listed								
				TA	TUA	TB	TUB	TD	DABC	DUB	DD	ACP31	ACP32	ACP53
Slicing	1S	6600.	1437938.	JUL	JUL	JUL	JUL	JUL	JUL	JUL	JUL	SEP	SEP	SEP
	2S	13200.	2875876.											
	3S	17160.	3738638.											
Lapping	1S	7990.	1794615.	JUL	JUL	JUL	JUL	JUL	JUL	JUL	JUL	SEP	SEP	SEP
	2S	15980.	3589229.											
	3S	20774.	4665998.											
Silicon Etch	1S	7460.	1782524.	AUG	JUL	JUL	JUL	JUL	JUL	JUL	JUL	SEP	SEP	SEP
	2S	14920.	3565048.											
	3S	19396.	4634563.											
Buffer Etch	1S	3762	1745176.	SEP	AUG	SEP	AUG	SEP	SEP	AUG	SEP	SEP	SEP	SEP
	2S	7524.	3490352.											
	3S	9781.	4537458.											
Aluminum Evap DABC	1S	2635.	1611846.	SEP	AUG	SEP	AUG	SEP	SEP	AUG	SEP	SEP	SEP	SEP
	2S	5270.	3223692.											
	3S	6851.	4190800.											
Fifth Appl	1S	5855.	1525553.	SEP	AUG	SEP	AUG	SEP	SEP	AUG	SEP	SEP	SEP	SEP
	2S	11710.	3051107.											
	3S	15223.	3966439.											

Fig. 2. Sample of program output.

- Production schedules

Contained within the program is all the information necessary to generate operational production schedules. Adding an additional output statement will provide this information.

- Cumulative yield data by product type

Contained within the program is all the information necessary to print out a cumulative yield matrix by product type. Adding an additional output statement will provide this information.

- Percentage of product through each operation

Contained within the program is all the information necessary to print out a matrix of the relative percentage of each product through each operation. Adding an additional output statement will provide this information.

DIMENSIONS OF THE PROGRAM

For purposes of evaluating this application to facilities of similar nature, it would be useful to discuss the dimensions of the variables contained within the program:

1. 155 discrete operations or less
2. 11 product types or less
3. a lead time of 5 time periods or less
4. an unlimited number of time periods per run

These dimensions may be altered, but this would require additional programming effort.

COMPILING DATA FOR USE IN THE PROGRAM

List all the operations being performed in the facility. These operations should then be amplified with respect to equipment rates. If equipment is isolated on the floor according to product or if the equipment runs at a different rate depending upon what product is being processed at an operation, break one operation into the number of operations required to cover these variables. Also, if the product cycles through an operation more than once, list it each time it cycles through the operation.

Using this information, make a listing of what product goes through each of the operations. When completed, this will serve as a routing of the flow of all products with respect to the sequence of operations and equipment rates. For example,

Product Flow (x denotes product going through an operation)

	Operation	1 TA	2 TUA	3 TB	4 TUB	5 TD	6 DABC	7 DUB	8 DD	9 ACP31	10 ACP52	11 ACP53
1.	Base Mask	x	x	x	x	x	x	x	x	x	x	x
2.	Boron diffusion TABC	x	x	x	x							
3.	Boron diffusion TD-ACP					x				x	x	x
4.	Boron diffusion DABCD						x	x	x			
5.	Second Alignment	x	x	x	x	x	x	x	x			
6.	Second Alignment ACP									x	x	x
7.	Third Alignment	x	x	x	x	x	x	x	x	x	x	x
8.	Third Alignment ACP									x	x	x

Boron diffusion was broken up into three operations since the equipment on the manufacturing floor is physically separated in the fashion shown above. Second and third alignment were separated between the ACPX product and the other product because the speed of the equipment is governed by this breakdown of product. Also, the same equipment is used for the second and third alignment (this will be discussed later).

A table based on the above operational flow and product routing is then prepared for inputing into the main computer program. Where the product is not going through an operation (an x does not appear next to the operation in the above example), a zero will be listed next to the operation number for each product type (refer to the card format section under Corr Input Appendix C.)

When product recycles through the same equipment, it is handled in the following manner: In the flow listing example, we saw that the second and third alignment was done on the same equipment. After statement number 94 in the main program listing, the following statements would be inserted to account for this:

```
capfd(5, 1) = capfd(5, 1)/(sumwtd(5)+sumwtd(7))*sumwtd(5)
capfd(7, 1) = capfd(5, 1)
```

Mathematically, this is allocating the capacity with respect to the throughput through these type operations.

APPENDIX A: FORTRAN PROGRAM LISTING

```
      DIMENSION CORR(155,11)
      DIMENSION IV(11),DAYS(6)
      DIMENSION PECOP(155,11),SUMWTD(155),WTDYLD(155,11),CUMYLD(155,11)
      DIMENSION CAPAT(155),CAPAT2(155),CAPAT3(155)
      DIMENSION MONTH(6),          CAPFD2(155,1),CAPFD3(155,1)
      DIMENSION OPER(155,3)
      DIMENSION MONO(155,11)
      DIMENSION ZOOM(155)
      DIMENSION IA(11),IB(11),IC(11),ID(11),IE(11),IS(11),IT(11),IU(11)
      DIMENSION L(155),CAPAX(155),TIMEX(155),YLDX(155)
       DIMENSION CONT(11,1),PCT(11,6)
      DIMENSION CHIP(11,1),CAPFD(155,1)
      EQUIVALENCE (WTDYLD(155,11),MONO(155,11))
      EQUIVALENCE (CUMYLD(155,11),PECOP(155,11))
      COMMON YLD(155,11,6),CAPAC(155,6)
      DO 690 I=1,155
      DO 690 J=1,11
  690 CORR(I,J)=1.0
      DO 685 J2=1,11
      READ (5,689) N,(L(M),M=1,N)
  689 FORMAT (3X,I3,22I3/6X,22I3/6X,22I3/6X,22I3)
      IF (IPASS.GT.4) GO TO 908
      DO 685 L1=1,N
      I=L(L1)
  685 CORR(I,J2)=0.
  908 CONTINUE
      DO 2000 IPASS=1,15
      IF (IPASS.EQ.1) GO TO 811
      READ (2)  (((YLD(I,J,K),I=1,155),J=1,11),K=1,6)
      REWIND 2
  811 CONTINUE
      CALL YIELDS (IPASS)
      REWIND 1
      REWIND 2
      WRITE (2)  ((( YLD(I,J,K),I=1,155),J=1,11),K=1,6)
      REWIND 2
      IF (IPASS.EQ.1) GO TO 891
      READ (1)  (((YLD(I,J,K),I=1,155),J=1,11),K=1,6)
      REWIND 1
  891 CONTINUE
      CALL CAPTIM (IPASS,OPER)
      WRITE (1)  (((YLD (I,J,K),I=1,155),J=1,11),K=1,6)
      REWIND 1
      K=1
      READ (5,503) (CHIP(J,K),J=1,11)
  503 FORMAT(11F5.0)
      READ (5,504) (CONT(J,K),J=1,11)
  504 FORMAT(11F5.0)
      IF (IPASS.NE.1) GO TO 75
       DO 569 K=1,6
  569  READ(5,505) (PCT(J,K),J=1,11)
  505  FORMAT(11F5.0)
      DO 511 K=1,6
      READ (5,506) MONTH(K),DAYS(K)
  506 FORMAT (10X,A6,4X,F5.0)
  511 CONTINUE
   75 CONTINUE
```

```
      IF(IPASS.EQ.1) GO TO 2005
       READ(5,505) (PCT(J,6),J=1,11)
      READ(5,506) MONTH(6),DAYS(6)
 2005 CONTINUE
      DO 2193 J=1,11
      IB(J)=0
      IC(J)=0
      ID(J)=0
      IE(J)=0
      IU(J)=0
      IV(J)=0
      IS(J)=0
      IT(J)=0
 2193 IA(J)=0
      K=5
      DO 29 J=1,11
      SUMA=0.
      DO 30 I=1,155
      I1=155-I
      SUMA=SUMA+YLD (I1,J,K)
      IF (J.NE.1) GO TO 22
      IF (I1.NE.19)GO TO 22
      IF (J.NE.1) GO TO 580
       KXX=K
  580 CONTINUE
   22 IF ((DAYS(K)/2.)-SUMA) 31,31,30
   30 IF (I1.EQ.1) GO TO 29
   31 IA(J)=I1
      IF (IA(J).EQ.1) GO TO 29
      IS(J)=IA(J)-1
      SUMB=0.
      DO 35 I=1,154
      I1=IA(J)-I
      SUMB=SUMB+YLD (I1,J,K-1)
      IF (J.NE.1) GO TO 23
      IF (I1.NE.19) GO TO 23
      IF (J.NE.1) GO TO 581
       KXX=K-1
  581 CONTINUE
   23 IF (DAYS(K-1)-SUMB) 99,36,35
   35 IF (I1.EQ.1) GO TO 29
   36 IB(J)=I1
      IF (IB(J).EQ.1) GO TO 29
      IT(J)=IB(J)-1
      SUMC=0.
      DO 40 I=1,155
      I1=IB(J)-I
      SUMC=SUMC+YLD (I1,J,K-2)
      IF (J.NE.1) GO TO 24
      IF (I1.NE.19) GO TO 24
      IF (J.NE.1) GO TO 582
       KXX=K-2
  582 CONTINUE
   24 IF (DAYS(K-2)-SUMC)99,41,40
   40 IF(I1.EQ.1) GO TO 29
   41 IC(J)=I1
      IF (IC(J).EQ.1) GO TO 29
      IU(J)=IC(J)-1
      SUMD=0.
      DO 45 I=1,155
      I1=IC(J)-I
      SUMD=SUMD+YLD (I1,J,K-3)
      IF (J.NE.1) GO TO 25
```

```
      IF (I1.NE.19) GO TO 25
      IF (J.NE.1) GO TO 583
       KXX=K-3
  583 CONTINUE
   25 IF (DAYS(K-3)-SUMD) 99,46,45
   45 IF (I1.EQ.1) GO TO 29
   46 ID(J)=I1
      IF (ID(J).EQ.1) GO TO 29
      IV(J)=ID(J)-1
      SUME=0.
      DO 50 I=1,155
      I1=ID(J)-I
      SUME=SUME+YLD (I1,J,K-4)
      IF (J.NE.1) GO TO 26
      IF (I1.NE.19) GO TO 26
      IF (J.NE.1) GO TO 584
       KXX=K-4
  584 CONTINUE
   26 IF (DAYS(K-4)-SUME) 99,51,50
   50 IF (I1.EQ.1) GO TO 29
   51 IE(J)=I1
      IF (IE(J).EQ.1) GO TO 29
   29 CONTINUE
      DO 1142 I=1,155
 1142 WRITE (6,1141) (OPER(I,M),M=1,3),(YLD(I,J,5),J=1,11)
 1141 FORMAT(2X,3A6,11F4.0)
      DO 711 J=1,11
      IF (IV(J).EQ.0)  IV(J)=1
      IF (IU(J).EQ.0)  IU(J)=1
      IF (IS(J).EQ.0)  IS(J)=1
      IF (IT(J).EQ.0)  IT(J)=1
      IF (IA(J).EQ.0)  IA(J)=2
      IF (IB(J).EQ.0)  IB(J)=2
      IF (IC(J).EQ.0)  IC(J)=2
      IF (ID(J).EQ.0)  ID(J)=2
      IF (IU(J).EQ.(-1))  IU(J)=1
      IF (IV(J).EQ.(-1))  IV(J)=1
      IF (IT(J).EQ.(-1))  IT(J)=1
      IF (IS(J).EQ.(-1)) IS(J)=1
  711 IF (IE(J).EQ.0)  IE(J)=1
      READ  (2)  ((( YLD(I,J,K),I=1,155),J=1,11),K=1,6)
      REWIND 2
      WRITE (2)  ((( YLD(I,J,K),I=1,155),J=1,11),K=1,6)
      REWIND 2
      DO 54 J=1,11
   54 CUMYLD (155,J) =1.
      K=5
      DO 55 J=1,11
      DO 56 I=1,154
      JA=155-I
      CUMYLD (JA,J)  =CUMYLD(JA+1,J)*1./YLD(JA,J,K)
      IF (JA.EQ.1)GO TO 55
      IF (IA(J).EQ.JA) GO TO 57
   56 CONTINUE
   57 IF (JA.EQ.1) GO TO 55
      IAJ=IA(J)
      DO 61 I=1,154
      JA=IA(J)-I
      CUMYLD(JA,J)    =CUMYLD(JA+1,J)*1. /YLD(JA,J,K-1)
      IF (IB(J).EQ.JA) GO TO 60
      IF (JA.EQ.1) GO TO 55
   61 CONTINUE
```

```
   60 IF (JA.EQ.1) GO TO 55
      IBJ=IB(J)
      DO 63 I=1,154
      JA=IB(J)-I
      CUMYLD(JA,J)     =CUMYLD(JA+1,J)*1. /YLD(JA,J,K-2)
      IF (IC(J).EQ.JA) GO TO 62
      IF (JA.EQ.1) GO TO 55
   63 CONTINUE
   62 IF (JA.EQ.1) GO TO 55
      ICJ=IC(J)
      DO 65 I=1,154
      JA=IC(J)-I
      CUMYLD(JA,J)     =CUMYLD(JA+1,J)*1. /YLD(JA,J,K-3)
      IF (ID(J).EQ.JA) GO TO 64
      IF (JA.EQ.1) GO TO 55
   65 CONTINUE
   64 IF (JA.EQ.1) GO TO 55
      IDJ=ID(J)
      DO 67 I=1,154
      JA=ID(J)-I
      CUMYLD(JA,J)     =CUMYLD(JA+1,J)*1. /YLD(JA,J,K-4)
      IF (IE(J).EQ.JA) GO TO 55
      IF (JA.EQ.1) GO TO 55
   67 CONTINUE
   55 CONTINUE
      DO 2102 I=1,155
 2102 WRITE (6,2101) (OPER(I,M),M=1,3),(CUMYLD(I,J),J=1,11)
 2101 FORMAT(2X,3A6,11F8.3)
      IF(IFOR.EQ.1) GO TO  1999
      KXX=5
      GO TO 2751
 1999 TOR=PCT(1,KXX)/CUMYLD(19,1)/CONT(1,1)+PCT(2,KXX)/CUMYLD(19,2)/CONT
     1(2,1)+PCT(3,KXX)/CUMYLD(19,3)/CONT(3,1)+PCT(4,KXX)/CUMYLD(19,4)/CO
     2NT(4,1)+PCT(5,KXX)/CUMYLD(19,5)/CONT(5,1)+PCT(6,KXX)/CUMYLD(19,6)/
     3CONT(6,1)+PCT(7,KXX)/CUMYLD(19,7)/CONT(7,1)+PCT(8,KXX)/CUMYLD(19,8
     4)/CONT(8,1)+PCT(9,KXX)/CUMYLD(19,9)/CONT(9,1)+PCT(10,KXX)/CUMYLD(1
     59,10)/CONT(10,1)+PCT(11,KXX)/CUMYLD(19,11)/CONT(11,1)
      PCT(1,KXX)=PCT(1,KXX)/CUMYLD(19,1)/CONT(1,1)/TOR
      PCT(2,KXX)=PCT(2,KXX)/CUMYLD(19,2)/CONT(2,1)/TOR
      PCT(3,KXX)=PCT(3,KXX)/CUMYLD(19,3)/CONT(3,1)/TOR
      PCT(4,KXX)=PCT(4,KXX)/CUMYLD(19,4)/CONT(4,1)/TOR
      PCT(5,KXX)=PCT(5,KXX)/CUMYLD(19,5)/CONT(5,1)/TOR
      PCT(6,KXX)=PCT(6,KXX)/CUMYLD(19,6)/CONT(6,1)/TOR
      PCT(7,KXX)=PCT(7,KXX)/CUMYLD(19,7)/CONT(7,1)/TOR
      PCT(8,KXX)=PCT(8,KXX)/CUMYLD(19,8)/CONT(8,1)/TOR
      PCT(9,KXX)=PCT(9,KXX)/CUMYLD(19,9)/CONT(9,1)/TOR
      PCT(10,KXX)=PCT(10,KXX)/CUMYLD(19,10)/CONT(10,1)/TOR
      PCT(11,KXX)=PCT(11,KXX)/CUMYLD(19,11)/CONT(11,1)/TOR
 2751 CONTINUE
      WRITE (6,2077) (PCT(J,5),J=1,11)
 2077 FORMAT (11F8.3)
      K=1
      DO 180 J=1,11
      DO 180 I=1,155
  180 WTDYLD(I,J)=CUMYLD(I,J)*PCT(J,5)/CONT(J,1)
      DO 71 J=1,11
      DO 72 I=1,155
      WTDYLD(I,J)=WTDYLD(I,J)*CORR(I,J)*1.
   72 CONTINUE
   71 CONTINUE
      DO 77 I=1,128
      SUMWTD(I)=WTDYLD(I,1)*1./CHIP(1,K)+WTDYLD(I,2)*1./CHIP(2,K)+WTDYL
```

```
  1D(I,3)*1./CHIP(3,K)+WTDYLD(I,4)*1./CHIP(4,K)+WTDYLD(I,5)*1./CHIP(
  25,K)+WTDYLD(I,6)*1./CHIP(6,K)+WTDYLD(I,7)*1./CHIP(7,K)+WTDYLD(I,8
  3)*1./CHIP(8,K)+WTDYLD(I,9)*1./CHIP(9,K)+WTDYLD(I,10)*1./CHIP(10,K
  4)+WTDYLD(I,11)*1./CHIP(11,K)
 77 CONTINUE
    DO 78 I=129,155
    SUMWTD(I)  =WTDYLD(I,1)             +WTDYLD(I,2)             +WTDYLD
  1(I,3)            +WTDYLD(I,4)            +WTDYLD(I,5)            +
  2WTDYLD(I,6)            +WTDYLD(I,7)            +WTDYLD(I,8)
  3     +WTDYLD(I,9)            +WTDYLD(I,10)             +WTDYLD(I,1
  41)
 78 CONTINUE
    K=5
    DO 85 I=1,155
    IN=155-I
    CAPFD(IN,1)=CAPAC(IN,K)/SUMWTD(IN)
    CAPAT(IN)=CAPAC(IN,K)
    IF (IN.EQ.1) GO TO 94
    IF (IA(1).EQ.IN) GO TO 86
 85 CONTINUE
 86 CONTINUE
    DO 87 I=1,155
    IN=IA(1)-I
    CAPFD(IN,1)=CAPAC(IN,K-1)/SUMWTD(IN)
    CAPAT(IN)=CAPAC(IN,K-1)
    IF (IN.EQ.1) GO TO 94
    IF (IB(1).EQ.IN) GO TO 88
 87 CONTINUE
 88 CONTINUE
    DO 89 I=1,155
    IN=IB(1)-I
    CAPFD(IN,1)=CAPAC(IN,K-2)/SUMWTD(IN)
    CAPAT(IN)=CAPAC(IN,K-2)
    IF (IN.EQ.1) GO TO 94
    IF (IC(1).EQ.IN) GO TO 90
 89 CONTINUE
 90 CONTINUE
    DO 91 I=1,155
    IN=IC(1)-I
    CAPFD(IN,1)=CAPAC(IN,K-3)/SUMWTD(IN)
    CAPAT(IN)=CAPAC(IN,K-3)
    IF (IN.EQ.1) GO TO 94
    IF (ID(1).EQ.IN) GO TO 92
 91 CONTINUE
 92 CONTINUE
    DO 93 I=1,155
    IN=ID(1)-I
    CAPFD(IN,1)=CAPAC(IN,K-4)/SUMWTD(IN)
    CAPAT(IN)=CAPAC(IN,K-4)
    IF(IN.EQ.1) GO TO 94
    IF (IE(1).EQ.IN) GO TO 94
 93 CONTINUE
 94 CONTINUE
    K=1
    DO 214 J=1,11
    DO 214 I=1,128
214 PECOP (I,J)  = ((WTDYLD(I,J)  /CHIP(J,K))/SUMWTD(I)  *1.)*100.
    DO 444 J=1,11
    DO 444 I=129,155
444 PECOP(I,J)  =(WTDYLD(I,J)  *1./SUMWTD(I)  )*100.
    WRITE (6,769) (PECOP(19,J),J=1,11)
    WRITE (6,769) (PECOP(155,J),J=1,11)
```

```
  769 FORMAT (11F12.6)
      K=5
       DO 222 J=1,11
       IM=IA(J)
       DO 215 I=IM,155
  215  MONO(I,J)=MONTH(K)
       IF (IS(J).EQ.1) GO TO 222
      MA=IS(J)
      MB=IB(J)
       DO  216 I=MB,MA
  216  MONO(I,J)=MONTH(K-1)
      MC=IT(J)
      MD=IC(J)
      IF(MC.EQ.1) GO TO 222
      DO 217 I=MD,MC
  217 MONO(I,J)=MONTH(K-2)
      ME=ID(J)
      MF=IU(J)
      IF(MF.EQ.1) GO TO 222
      DO 218 I=ME,MF
  218 MONO(I,J)=MONTH(K-3)
      MG=IE(J)
      MH=IV(J)
      IF(MH.EQ.1) GO TO 222
      DO 219 I=MG,MH
  219 MONO(I,J)=MONTH(K-4)
  222 CONTINUE
      DO 221 I=1,155
      CAPFD2(I,1)=CAPFD(I,1)*2.
      CAPFD3(I,1)=CAPFD(I,1)*2.6
      CAPAT2(I)=CAPAT(I)*2.
  221 CAPAT3(I)=CAPAT(I)*2.6
      K=5
      IF (IPASS.LT.6) GO TO 1777
      WRITE (6,226) MONTH(K)
  226 FORMAT (50X,A6,2X,17HOUTPUT  TO  STOCK/54X,10HEQUIVALENT/13X,14HOP
     1ERATION NAME,6X,7HPLANNED,14X,10HFIN DEVICE,13X,44HPRODUCT IMPACTS
     2 CAPACITY DURING MONTH LISTED/32X,8HCAPACITY,15X,8HCAPACITY,2X,65H
     3   TA TA-ULD  TB  TB-ULD  TD   DABC DB-ULD  DD  ACP3B ACP52 ACP53)
      DO 225I=1,155
      WRITE (6,812)          (OPER(I,M),M=1,3),CAPAT(I)  ,CAPFD(I,1),MONO
     1(I,1),MONO(I,2),MONO(I,3),MONO(I,4),MONO(I,5),MONO(I,6),MONO(I,7),
     2MONO(I,8),MONO(I,9),MONO(I,10),MONO(I,11),CAPAT2(I)  ,CAPFD2(I,1),
     3CAPAT3(I)  ,CAPFD3(I,1)
  225 CONTINUE
  812 FORMAT      (11X,3A6,2H1S,F10.0,13X,F10.0,11A6/29X,2H2S,F10.0,13X,
     1F10.0/29X,2H3S,F10.0,13X,F10.0/)
 1777 CONTINUE
   99 CONTINUE
      DO 74 K=2,6
       DO 74 J=1,11
       PCT(J  ,K-1)=PCT(J,K)
      MONTH(K-1)=MONTH(K)
   74 DAYS(K-1)=DAYS(K)
 2000 CONTINUE
      STOP
      END
$IBFTC YIELDS
      SUBROUTINE YIELDS(IPASS)
      DIMENSION YLDX(155),L(155),TIMEX(155),CAPAX(155),OPER(155,3)
      COMMON YLD(155,11,6),CAPAC(155,6)
      IF (IPASS.EQ.1) GO TO 40
```

```
      DO 311 I=1,155
      DO 311 J=1,11
      DO 311 K=2,6
  311 YLD (I,J,K-1)=YLD (I,J,K)
   40 CONTINUE
      IF (IPASS.NE.1) GO TO 515
      DO 500 I=1,155
  500 READ (5,601) (YLD(I,J,1),J=1,11)
  601 FORMAT (15X,11F5.0)
      DO 502 I=1,155
      DO 502 J=1,11
  502 YLD (I,J,2)=YLD(I,J,1)
      DO 501 J1=1,11
      READ (5,602) N,(L(M),YLDX(M),M=1,N)
      WRITE (6,602) N,(L(M),YLDX(M),M=1,N)
      DO 501 J=1,N
      I=L(J)
  501 YLD(I,J1,2)=YLDX(J)
      DO 503 J=1,11
      DO 503 I=1,155
  503 YLD (I,J,3)=YLD(I,J,2)
      DO 504 J1=1,11
      READ (5,602) N,(L(M),YLDX(M),M=1,N)
      WRITE (6,602) N,(L(M),YLDX(M),M=1,N)
      DO 504 J=1,N
      I=L(J)
  504 YLD (I,J1,3)=YLDX(J)
      DO 505 I=1,155
      DO 505 J=1,11
  505 YLD (I,J,4)=YLD(I,J,3)
      DO 506 J1=1,11
      READ (5,602) N,(L(M),YLDX(M),M=1,N)
      WRITE (6,602) N,(L(M),YLDX(M),M=1,N)
      DO 506 J=1,N
      I=L(J)
  506 YLD (I,J1,4)=YLDX(J)
      DO 507 I=1,155
      DO 507 J=1,11
  507 YLD (I,J,5)=YLD(I,J,4)
      DO 508 J1=1,11
      READ (5,602) N,(L(M),YLDX(M),M=1,N)
      WRITE (6,602) N,(L(M),YLDX(M),M=1,N)
      DO 508 J=1,N
      I=L(J)
  508 YLD (I,J1,5)=YLDX(J)
      DO 509 J=1,11
      DO 509 I=1,155
  509 YLD (I,J,6)=YLD(I,J,5)
  515 CONTINUE
      DO 510 J1=1,11
      READ (5,602) N,(L(M),YLDX(M),M=1,N)
      WRITE (6,602) N,(L(M),YLDX(M),M=1,N)
      DO 510 J=1,N
      I=L(J)
  510 YLD(I,J1,6)=YLDX(J)
  602 FORMAT(6X,I3,8(I3,F5.0)/9X,8(I3,F5.0)/9X,8(I3,F5.0)/9X,8(I3,F5.0)/
     19X,8(I3,F5.0))
      RETURN
      END
$IBFTC CAPTIM
      SUBROUTINE  CAPTIM (IPASS,OPER)
      DIMENSION YLDX(155),L(155),TIMEX(155),CAPAX(155),OPER(155,3)
      COMMON YLD(155,11,6),CAPAC(155,6)
```

```
      IF (IPASS.EQ.1) GO TO 50
      DO 531 I=1,155
      DO 531 K=2,6
  531 CAPAC(I,K-1)=CAPAC(I,K)
      DO 511 I=1,155
      DO 511 J=1,11
      DO 511 K=2,6
  511 YLD (I,J,K-1)=YLD (I,J,K)
   50 CONTINUE
      IF (IPASS.NE.1) GO TO 515
      DO 500 I=1,155
  500 READ(5,601) (OPER(I,M),M=1,3),(YLD (I,J,1),J=1,11),CAPAC(I,1)
  601 FORMAT(2X,3A6,11F2.0,8X,F10.0)
      DO 502 I=1,155
      DO 502 J=1,11
  502 YLD (I,J,2)=YLD (I,J,1)
      DO 501 J1=1,11
      READ(5,602) N,(L(M),TIMEX(M),M=1,N)
      WRITE (6,602) N,(L(M),TIMEX(M),M=1,N)
      DO 501 J=1,N
      I=L(J)
  501 YLD (I,J1,2)=TIMEX(J)
      DO 503 J=1,11
      DO 503 I=1,155
  503 YLD (I,J,3)=YLD (I,J,2)
      DO 504 J1=1,11
      READ(5,602) N,(L(M),TIMEX(M),M=1,N)
      WRITE (6,602) N,(L(M),TIMEX(M),M=1,N)
      DO 504 J=1,N
      I=L(J)
  504 YLD (I,J1,3)=TIMEX(J)
      DO 505 I=1,155
      DO 505 J=1,11
  505 YLD (I,J,4)=YLD (I,J,3)
      DO 506 J1=1,11
      READ(5,602) N,(L(M),TIMEX(M),M=1,N)
      WRITE (6,602) N,(L(M),TIMEX(M),M=1,N)
      DO 506 J=1,N
      I=L(J)
  506 YLD (I,J1,4)=TIMEX(J)
      DO 507 I=1,155
      DO 507 J=1,11
  507 YLD (I,J,5)=YLD (I,J,4)
      DO 508 J1=1,11
      READ(5,602) N,(L(M),TIMEX(M),M=1,N)
      WRITE (6,602) N,(L(M),TIMEX(M),M=1,N)
      DO 508 J=1,N
      I=L(J)
  508 YLD (I,J1,5)=TIMEX(J)
      DO 509 J=1,11
      DO 509 I=1,155
  509 YLD (I,J,6)=YLD (I,J,5)
  515 CONTINUE
      DO 510 J1=1,11
      READ(5,602) N,(L(M),TIMEX(M),M=1,N)
      WRITE (6,602) N,(L(M),TIMEX(M),M=1,N)
      DO 510 J=1,N
      I=L(J)
  510 YLD (I,J1,6)=TIMEX(J)
  602 FORMAT(6X,I3,12(I3,F2.0)/9X,12(I3,F2.0)/9X,12(I3,F2.0))
      IF(IPASS.NE.1)GO TO 535
      DO 520 I=1,155
```

```
  520 CAPAC(I,2)=CAPAC(I,1)
      READ(5,612) N,(L(M),CAPAX(M),M=1,N)
      WRITE (6,612) N,(L(M),CAPAX(M),M=1,N)
      DO 521 J=1,N
      I=L(J)
  521 CAPAC(I,2)=CAPAX(J)
      DO 523 I=1,155
  523 CAPAC(I,3)=CAPAC(I,2)
      READ(5,612) N,(L(M),CAPAX(M),M=1,N)
      WRITE (6,612) N,(L(M),CAPAX(M),M=1,N)
      DO 524 J=1,N
      I=L(J)
  524 CAPAC(I,3)=CAPAX(J)
      DO 525 I=1,155
  525 CAPAC(I,4)=CAPAC(I,3)
      READ(5,612) N,(L(M),CAPAX(M),M=1,N)
      WRITE (6,612) N,(L(M),CAPAX(M),M=1,N)
      DO 526 J=1,N
      I=L(J)
  526 CAPAC(I,4)=CAPAX(J)
      DO 527 I=1,155
  527 CAPAC(I,5)=CAPAC(I,4)
      READ(5,612) N,(L(M),CAPAX(M),M=1,N)
      WRITE (6,612) N,(L(M),CAPAX(M),M=1,N)
      DO 528 J=1,N
      I=L(J)
  528 CAPAC(I,5)=CAPAX(J)
      DO 529 I=1,155
  529 CAPAC(I,6)=CAPAC(I,5)
  535 CONTINUE
      READ(5,612) N,(L(M),CAPAX(M),M=1,N)
      WRITE (6,612) N,(L(M),CAPAX(M),M=1,N)
      DO 530 J=1,N
      I=L(J)
  530 CAPAC(I,6)=CAPAX(J)
  612 FORMAT(3X,I3,5(I3,F10.0)/6X,5(I3,F10.0)/6X,5(I3,F10.0)/6X,5(I3,F10
     1.0)/6X,5(I3,F10.0)/6X,5(I3,F10.0)/6X,5(I3,F10.0)/6X,5(I3,F10.0)/6X
     2,5(I3,F10.0)/6X,5(I3,F10.0)/6X,5(I3,F10.0))
      RETURN
      END
$ENTRY
```

APPENDIX B : DATA FORMATS (See Fig. 3 for Stacking of Data)

CHIPS PER WAFER

Column Number	Column Entry
1 - 5	Chips on TA wafer (floating points)
6-10	Chips on TUA wafer (floating point)
11-15	Chips on TB wafer (floating point)
16-20	Chips on TUB wafer (floating point)
21-25	Chips on TD wafer (floating point)
26-30	Chips on DB wafer (floating point)
31-35	Chips on DUB wafer (floating point)
36-40	Chips on DD wafer (floating point)
41-45	Chips on TF31 wafer (floating point)
46-50	Chips on TF52 wafer (floating point)
51-55	Chips on TF53 wafer (floating point)
56-80	Identification of time period (not picked up)

Note: One card per time period.

CONTINGENCY

Column Number	Column Entry
1-5	Contingency for TA yields (foating point)
6-10	Contingency for TUA yields (floating point)
11-15	Contihgency for TB yields (floating point)
16-20	Contingency for TUB yields (floating point)
21-25	Contingency for TD yields (floating point)
26-30	Contingency for DB yields (floating point)
31-35	Contingency for DUB yields (floating point)
36-40	Contingency for DD yields (floating point)
41-45	Contingency for TF31 yields (floating point)
46-50	Contingency for TF52 yields (floating point)
51-55	Contingency for TF53 yields (floating point)
56-80	Identification of time period (not picked up)

Note: One card per time period.

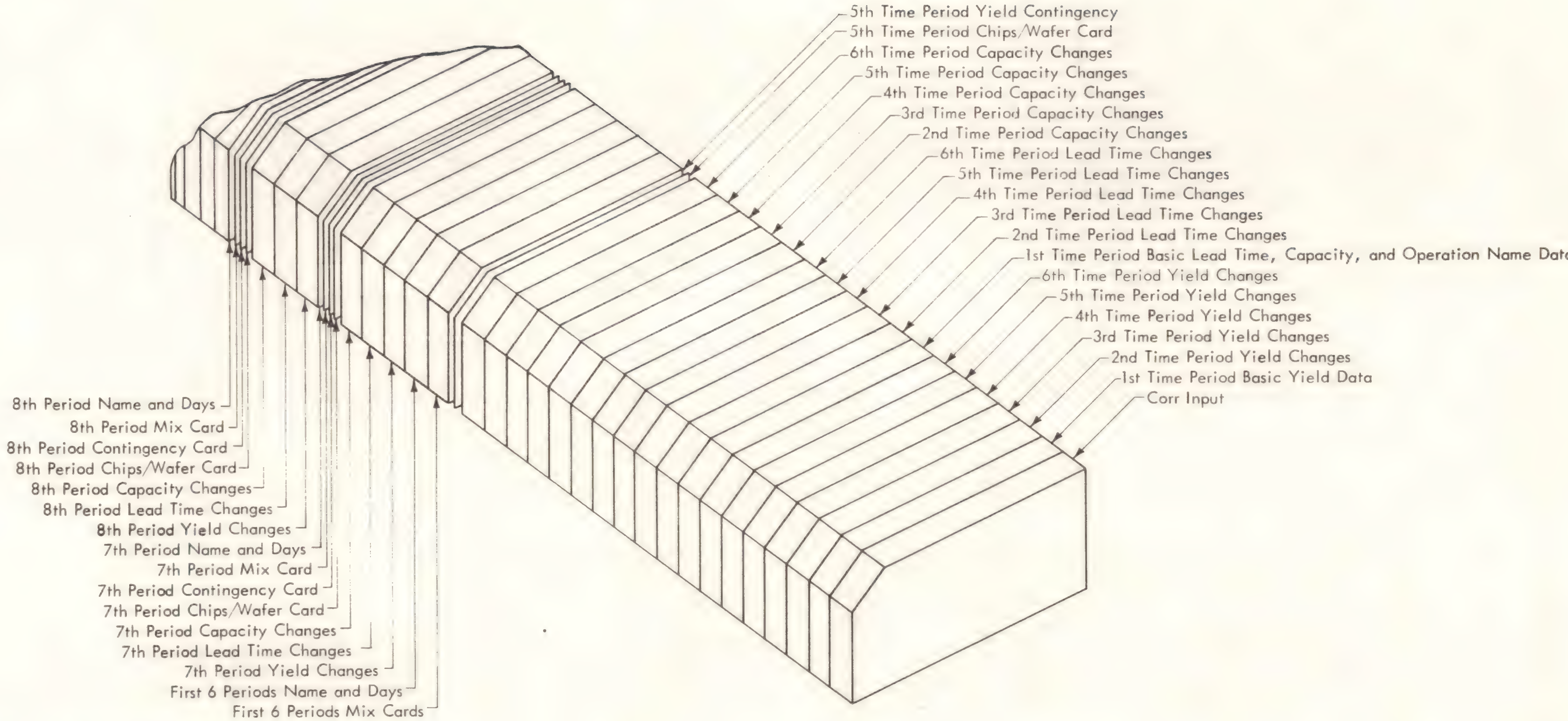

Fig. 3. Stacking of data.

MIX

Column Number	Column Entry
1-5	Base Mask TA percentage (floating point)
6-10	Base Mask TUA percentage (floating point)
11-15	Base Mask TB percentage (floating point)
16-20	Base Mask TUB percentage (floating point)
21-25	Base Mask TD percentage (floating point)
26-30	Base Mask DB percentage (floating point)
31-35	Base Mask DUB percentage (floating point)
36-40	Base Mask DD percentage (floating point)
41-45	Base Mask TF31 percentage (floating point)
46-50	Base Mask TF52 percentage (floating point)
51-55	Base Mask TF53 percentage (floating point)
56-80	Identification of time period (not picked up)

Note: One card per time period.

TIME PERIODS AND DAYS

Column Number	Column Entry
1-10	Blank
11-16	Name of time period (letters)
17-20	Blank
21-25	Days in time period (floating point)

Note: One card per time period.

BASIC YIELD TABLE

Column Number	Column Entry
1-15	Identification of operation (not picked up)
16-20	TA yield (floating point)
21-25	TUA yield (floating point)
26-30	TB yield (floating point)
31-35	TUB yield (floating point)
36-40	TD yield (floating point)
41-45	DABC yield (floating point)
46-50	DUB yield (floating point)
51-55	DD yield (floating point)
56-60	TF31 yield (floating point)

61-65	TF52 yield (floating point)
66-70	TF53 yield (floating point)

Note: Number of cards (operations) = 155

CHANGES TO YIELDS

Column Number	Column Entry
1-6	Blank
7-9	Number of changes of yield (fixed point)
10-12	Operation number (fixed point)
13-17	Yield change (floating point)
18-20	Operation Number (fixed point)
21-25	Yield change (floating point)
26-28	Operation Number (fixed point)
29-33	Yield change (floating point)
34-36	Operation number (fixed point)
37-41	Yield change (floating point)
42-44	Operation number (fixed point)
45-49	Yield change (floating point)
50-52	Operation number (fixed point)
53-57	Yield change (floating point)
58-60	Operation number (fixed point)
61-65	Yield change (floating point)
66-68	Operation number (fixed point)
69-73	Yield change (floating point)
74-80	Month and type identification (not picked up)

Notes: 1. Up to five cards per type (for second card on start in column 10). If last change comes out on column 69-73, repeat on next card.

2. Eleven types

BASIC LEAD TIME, OPERATION NAME, CAPACITY TABLE

Column Number	Column Entry
1-2	Blank
3-20	Operation name (letters)

21-22	TA lead time (floating point)
23-23	TUA lead time (floating point)
25-26	TB lead time (floating point)
27-28	TUB lead time (floating point)
29-30	TD lead time (floating point)
31-32	DB lead time (floating point)
33-34	DUB lead time (floating point)
35-36	DD lead time (floating point)
37-38	TF31 lead time (floating point)
39-40	TF52 lead time (floating point)
41-42	TF53 lead time (floating point)
43-50	Blank
51-60	Capacity (floating point)

Note: Number of cards (operations) = 155.

CHANGES TO LEAD TIMES

Column Number	Column Entry
1-6	Blank
7-9	Number of changes to lead times (fixed point)
10-12	Operation number (fixed point)
13-14	Lead time change (floating point)
15-17	Operation number (fixed point)
18-19	Lead time change (floating point)
20-22	Operation number (fixed point)
23-24	Lead time change (floating point)
25-27	Operation number (fixed point)
28-29	Lead time change (floating point)
30-32	Operation number (fixed point)
33-34	Lead time change (floating point)
35-37	Operation number (fixed point)
38-39	Lead time change (floating point)
40-42	Operation number (fixed point)
43-44	Lead time change (floating point)
45-47	Operation number (fixed point)
48-49	Lead time change (floating point)

50-52	Operation number (fixed point)
53-54	Lead time change (floating point)
55-57	Operation number (fixed point)
58-59	Lead time change (floating point)
60-62	Operation number (fixed point)
63-64	Lead time change (floating point)
65-67	Operation number (fixed point)
68-69	Lead time change (floating point)
71-80	Month and type identification (not picked up)

Notes: 1. Up to three cards per type (for second card on start in column 10). If last change comes out on columns 68-69, repeat on next card.

2. Eleven types.

CHANGES TO CAPACITIES
(OR THROUGHOUTS)

Column Number	Column Entry
1-3	Blank
4-6	Number of changes to capacities or throughputs (fixed point)
7-9	Operation number (fixed point)
10-19	Capacity change (floating point)
20-22	Operation number (fixed point)
23-32	Capacity change (floating point)
33-35	Operation number (fixed point)
36-45	Capacity change (floating point)
46-48	Operation number (fixed point)
49-58	Capacity change (floating point)
59-61	Operation number (fixed point)
62-71	Capacity change (floating point)
72-80	Month identification (not picked up)

Note: Up to eleven cards (for second card start in column 7). If last change comes out on columns 62-71, repeat on next card.

APPENDIX C : CORR INPUTS

Column Number	Operation #	
1-3	blank	
6-9	# of insertions	
10-12	Operation #	(Fixed Point)
13-15		
16-18		
19-21		
22-24		
25-27		
28-30		
31-33		
34-36		
37-39		
40-42		
43-45		
46-48		
49-51		
52-54		
55-57		
58-60		
61-63		
64-66		
67-69		
70-72		
73-75		

Notes: 1. Up to 4 cards per type (for second card start in column 7). If last entry comes out on columns 73-75, repeat on next card.

2. Eleven types.